BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

[redacted] CSS 06043

SUBJECT: A System Contraction Program (SYCO) For Structural Dynamics - Case 320 (Revision)

DATE: June 13, 1969

FROM: S. N. Hou

ABSTRACT

A program for contracting the sizes of mass and stiffness matrices with up to 300 degrees-of-freedom (DOF) from a system with up to 600 DOF is proposed. For a system having more than 600 DOF, successive contractions are recommended with not more than 300 DOF being reduced each cycle. The final retained DOF will be less than 300, which is the limit of our Modal Analysis Program (MAP).

Out-core-storage (OCS) and data overlay techniques are suggested for achieving an efficient use of the computer in-core-storage space, a shorter program running time, and a larger program capability in DOF. Certain capabilities, such as input mass and stiffness matrices, inverse of a positive definite symmetric matrix by Cholesky decomposition, and product of matrices, have been developed in the MAP program using OCS for systems with up to 300 DOF. Hence, the System Contraction Program can readily be developed from existing subroutines available from the MAP program.

Analytical as well as programming procedures are presented. The handling of data storage throughout the whole computation scheme is emphysized, which is essential for such a sizable amount of data.

1969 NOV RECEIVED NASA STI FACILITY INPUT BRANCH

[redacted]

(NASA-CR-106447) A SYSTEM CONTRACTION PROGRAM /SYCO/ FOR STRUCTURAL DYNAMICS (Bellcomm, Inc.) 9 p

N79-73193

Unclas
00/66 11559

NASA CORS CENTRAL FILES

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

69 C6048

SUBJECT: A System Contraction Program (SYCO) For Structural Dynamics - Case 320

DATE: June 13, 1969

FROM: S. N. Hou

MEMORANDUM FOR FILE

I. INTRODUCTION:

To achieve a high fidelity in modeling the structure of a space vehicle by an equivalent discrete system, a great number of nodal points must be taken. Thus, the number of degrees-of-freedom (DOF) involved may exceed the available capability for computation. In order to make dynamic analysis feasible, special treatment for reducing the size of problems is needed. According to the state-of-art, at least two approaches may be used:

A. Modal Synthesis---Partition a system to several subsystems and perform modal analysis for each subsystem. The modal data of the whole system may be computed by synthesizing certain selected modes from each subsystem.

B. System Contraction---Set up an equivalent new system which contains only a selected portion of its original DOF, but will yield the modal information as if the deleted portion remains.

In the following, we will focus on the latter topic of system contractions, explain its theoretical aspect, and propose a programming scheme suitable for a large system.

II. ANALYTICAL EXPLANATION:

For a linear discrete system under free vibration, the undamped equations of motion can be expressed in matrix form as follows:

$$[M] \{\ddot{U}\} + [K] \{U\} = \{0\}, \qquad (1)$$

where: $[M]$ = Mass matrix,
$[K]$ = Stiffness matrix,
$\{U\}$ = Displacement vector.

Notice that $[M]$ and $[K]$ are invariant and suitable for all possibles $\{U\}$'s. Thus, if the contracted system remain linear, we can find a contracted mass matrix $[M_c]$ an stiffness matrix

$[K_c]$ set, for a particular displacement vector $[U_c]$. Then the same $[M_c]$ and $[k_c]$ are likely suitable for all other possible expressions of $[U_c]$'s.

For a system under rigid body motion, the following relations are true:

$$[K]\{U\} = 0. \tag{2}$$

Now, lets designate

$\{U_1\}$ = Displacement vector containing all selected DOF for the contracted system, and

$\{U_2\}$ = Displacement vector containing all DOF to be deleted.

Thus eq. (2) is partitioned as:

$$\left[\begin{array}{c|c} K_{11} & K_{12} \\ \hline K_{21} & K_{22} \end{array}\right] \left\{\begin{array}{c} U_1 \\ \hline U_2 \end{array}\right\} = \left\{\begin{array}{c} 0 \\ \hline 0 \end{array}\right\} \tag{3}$$

which means,

$$[K_{11}]\{U_1\} + [K_{12}]\{U_2\} = \{0\} \tag{4}$$

$$[K_{21}]\{U_1\} + [K_{22}]\{U_2\} = \{0\}. \tag{5}$$

From eq. (5), we can express

$$\{U\} = \left\{\begin{array}{c} U_1 \\ U_2 \end{array}\right\} = \left[\begin{array}{c} I \\ \hline -K_{22}^{-1} K_{21} \end{array}\right] \{U_1\} \tag{6}$$

or

$$\{U\} = [T]\{U_1\} \tag{7}$$

by defining

$$[T] = \left[\begin{array}{c} I \\ \hline -K_{22}^{-1} K_{21} \end{array}\right]. \tag{8}$$

Then, substitute eq. (6) into eq. (4), we have

$$\left(\left[K_{11} \vdots K_{12} \right] \left[\begin{array}{c} I \\ \hline -K_{22}^{-1} K_{21} \end{array} \right] \right) \left\{ U_1 \right\} = \left\{ 0 \right\}, \qquad (9)$$

which is an expression similar to eq. (2) and may be considered as the contracted system under rigid body motion. Thus, the new stiffness matrix for the contracted system is:

$$\left[K_c \right] = \left[K_{11} \vdots K_{12} \right] \left[\begin{array}{c} I \\ \hline -K_{22}^{-1} K_{21} \end{array} \right], \qquad (10)$$

and,

$$\left[K_c \right] = \left[K_{11} \right] - \left[K_{12} \right] \left[K_{22} \right]^{-1} \left[K_{21} \right]. \qquad (11)$$

This is actually a result of the following transformation:

$$\left[K_c \right] = [T]^T [K] [T]. \qquad (12)$$

The same transformation will yield the mass matrix of the contracted system as

$$\left[M_c \right] = [T]^T [M] [T]. \qquad (13)$$

if [M] is partitioned in the same fashion as [K].

The same results will be obtained if we substitute eq. (7) directly into eq. (1) as

$$[M] [T] \left\{ \ddot{U}_1 \right\} + [K] [T] \left\{ U_1 \right\} = \left\{ 0 \right\}, \qquad (14)$$

and then restore symmetry of matrices by pre-multiplying by $[T]^T$:

$$\left([T]^T [M] [T] \right) \left\{ \ddot{U}_1 \right\} + \left([T]^T [K] [T] \right) \left\{ U_1 \right\} = \{0\}. \qquad (15)$$

We also get

$$\left[M_c\right] = [T]^T [M] [T]$$

$$= \left[\begin{array}{c|c} I & -K_{22}^{-1} K_{21} \end{array}\right] \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] \left[\begin{array}{c} I \\ \hline -K_{22}^{-1} K_{21} \end{array}\right] \tag{16}$$

and

$$\left[K_c\right] = [T]^T [K] [T]$$

$$= \left[\begin{array}{c|c} I & -K_{22}^{-1} K_{21} \end{array}\right] \left[\begin{array}{c|c} K_{11} & K_{12} \\ \hline K_{21} & K_{22} \end{array}\right] \left[\begin{array}{c} I \\ \hline -K_{22}^{-1} K_{21} \end{array}\right]$$

$$= \left[K_{11}\right] - \left[K_{12}\right] \left[K_{22}\right]^{-1} \left[K_{21}\right]. \tag{17}$$

They are identical with expressions in eqs. (13) and (11).

III. PROGRAMMING LOGIC.

Thus, major steps involved in SYCO programming are proposed in successive order as follows:

A. Input Data:

1. Read in mass and stiffness matrice and store all their elements in proper sequence.

2. Specify DOF's which are intended to remain in the contracted system (or specify DOF's to be deleted).

B. Computation Routines:

Since matrices [M] or [K] are contracted in the

same manner, the following scheme can be used repeatedly:

1. Rearrange rows and columns of matrix [M] or [K] according to DOF's specified in Sec. III. A. 2., and partition the matrix as shown in eq. (3).

2. Inverse matrix $\left[K_{22}\right]^{-1}$.

3. Compute transform matrix as

$$[T] = \begin{bmatrix} I \\ \hline -K_{22}^{-1} K_{21} \end{bmatrix} \tag{18}$$

4. Obtain contracted mass and stiffness matrices as

$$\left[M_c\right] = [T]^T [M] [T] \tag{19}$$

and

$$\left[K_c\right] = [T]^T [K] [T] \tag{20}$$

respectively.

C. Output contracted matrices.

IV. PROGRAMMING SHORTCUT VIA 'MAP':

In order to optimize programming time and avoid duplication of efforts, certain program routines, which have already been developed or will soon be completed in other programs, and which may also fit this program, are certainly worth our consideration. The following portions in Program 'MAP' may be adopted in this program.

A. Input data, (especially read in Boeing's mass and stiffness matrices), and store them in OCS in proper sequence.

B. Inverse a symmetric positive definite matrix [M] by decomposing it into a product of a lower triangular matrix [L] and its transpose matrix $[L]^T$,

$$[M] = [L] [L]^T \tag{21}$$

and then find

$$[M]^{-1} = [L]^{-1^T} [L]^{-1}. \qquad (22)$$

C. Compute matrix product.

Thus the actual portions needed to be programmed are very minor. The only portion requiring an explicit explanation in programming detail is rearranging rows and columns, and partitioning the matrix, (Sec. III. B. 1), which is a relatively simple matter.

V. PARTITIONING SCHEME:

Herewith, a programming scheme is described for matrix partitioning, which is the only portion which needs programming effort. As we know, for dealing a big system with limited core storage, we have to store input data in OCS, bring in portions of data at a time from OCS for computation, and then storing the partial results back to OCS. Now, let's take matrix [M] for description purposes, and understanding that matrix [K] will be handled in the same fashion:

A. Suppose the [M] already has been read in and stored in a one dimensional OCS location 1, column by column. (Since [M] is symmetric, the sequence is the same as row by row).

B. Bring in core a maximum number of columns of [M] at a time from OCS location 1. The number is restricted by the available core space assigned for it.

C. For each column of [M], check whether its DOF is specified to be retained in the contracted systems, until all columns are checked:

1. If it is yes, store those elements in certain rows, which have their DOF specified to be retained in the contracted systems, in OCS location 2. The remaining elements are stored in OCS location 3.

2. If it is no, store those elements in certain rows, which have their DOF specified to be retained, in OCS location 4 and those specified to be deleted elements in OCS location 5.

Thus, we complete the row and column exchanges, and

also complete the partitioning of matrix [M] in the following way:

A. $[M_{11}]$ is stored column by column (same as row by row due to symmetry) in OCS location 2.

B. $[M_{22}]$ is stored column by column (same as row by row due to symmetry) in OCS location 5.

C. $[M_{21}]$ is stored column by column in OCS location 3, which is equivalent to storing $[M_{12}]$ row by row, since

D. $[M_{12}]$ is stored column by column in OCS location 4.

Then we put these four portions back to a unique matrix [M] by storing elements column by column in a one dimension OCS new location. This is the partitioned matrix we need for further operations.

VI. CONCLUSION AND DISCUSSION:

Up to the present, we have the modal analysis program MAP planned for only 300 DOF. However, we already foresee structural models of space vehicles having DOF larger than 300. The need of systems contraction program SYCO is evident.

As previously mentioned, the logic for such a program is straight forward. Especially, major portions of programming have already been developed in the previous program MAP. Thus, a minimum effort may provide us a tool for dealing with bigger systems.

Since 300 DOF is the maximum capability for modal analysis, we would always like a larger system to be reduced to a size equal to or less than 300 DOF. Since we plan to use matrix inversion routine from program MAP, which can handle 300 DOF at most, we can only reduce 300 DOF or less in each contraction operation. This is the primary limitation of the SYCO program.

Since matrix inversion is involved, $[K_{22}]$ must be non-singular. The program user should be aware of this in choosing which DOF's are to be deleted.

2031-SNH-ekh

S. N. Hou

BELLCOMM, INC.

SUBJECT: A System Contraction Program (SYCO) for Structural Dynamics Case 320

FROM: S. N. Hou

DISTRIBUTION

Bellcomm, Inc.

G. M. Anderson
C. Bidgood
A. P. Boysen, Jr.
D. R. Hagner
H. A. Helm
J. J. Hibbert
N. W. Hinners
W. W. Hough
B. T. Howard
D. B. James
R. K. McFarland
J. Z. Menard
J. J. O'Connor
I. M. Ross
J. A. Saxton
R. V. Sperry
J. W. Timko
R. L. Wagner
Central Files
Department 1024 File
Department 2031
Library